# Macros

# Macros

Write code that writes code - meta programming

# Macros

Write code that writes code - meta programming

Match an expression and perform some operation

# Macros

Write code that writes code - meta programming

Match an expression and perform some operation

Code is expanded and compiled

# Macros

Write code that writes code - meta programming

Match an expression and perform some operation

Code is expanded and compiled

```
macro_rules! my_macro {
   (match) => ( code to run )
}
```

# Macros

Write code that writes code - meta programming

Match an expression and perform some operation

Code is expanded and compiled

```
macro_rules! my_macro {
  (match) => ( code to run )
}
```

```
my_macro!
```

# Macros

Write code that writes code - meta programming

Match an expression and perform some operation

Code is expanded and compiled

```
macro_rules! my_macro {
  (match) => ( code to run )
}
```

```
my_macro!
```

```
println!(“This is an {} macro”, “awesome”);
```

# Macros

## We can match multiple expressions

```
macro_rules! my_macro {
  (match1) => ( code to run )
  (match2) => ( code to run )
}
```

# Macros

## We can match multiple expressions

```
macro_rules! my_macro {
  (match1) => ( code to run )
  (match2) => ( code to run )
}
```

## Designators

- expr
- ident
- block
- stmt
- pat
- path
- meta
- ty
- tt